**ПРОГРАММИРОВАНИЕ ВЫЧИСЛИТЕЛЬНЫХ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ПОДПРОГРАММ ПОЛЬЗОВАТЕЛЯ.**

[ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ 1](#_Toc252111217)

[КОНТРОЛЬНЫЕ ВОПРОСЫ 4](#_Toc252111218)

[ВАРИАНТЫ ЗАДАНИЙ 4](#_Toc252111219)

# ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Любая задача решается гораздо легче, если разбить ее на части - процедуры и функции. Например, задача по созданию школьной базы данных разбивается на три процедуры: создание БД, использование данных БД, корректировка БД. В пользовательские процедуры и функции можно передавать значения и после выполнения получать результат. Сначала рассмотрим процедуры.

ПРОЦЕДУРЫ ПОЛЬЗОВАТЕЛЯ

Процедура начинается словом PROCEDURE, по структуре аналогична программе:

PROCEDURE <имя> (необязательные формальные параметры);

LABEL ...; CONST ...; TYPE ...; VAR ...;

BEGIN

<операторы>

END;

Для выполнения процедуры достаточно указать ее имя и, если они нужны, передаваемые в процедуру данные (фактические параметры), совпадающие по типу и порядку следования с требуемыми (формальными параметрами).

ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ

Функция начинается словом FUNCTION, за которым следует имя, необязательный список формальных параметров и тип возвращаемого результата:

FUNCTION <имя> (Необязательные формальные параметры) : <тип результата>;

 LABEL ...; CONST ...; TYPE ...; VAR ...;

 BEGIN

 <операторы>

 END;

В разделе операторов функции должен находиться по крайней мере один оператор, присваивающий имени функции значение. В точку вызова возвращается результат последнего присваивания.

ПАРАМЕТРЫ

Если список параметров начинается словом VAR, то они передаются из процедуры после ее выполнения и результаты присваиваются соответствующим фактическим параметрам. Если при перечислении формальных параметров VAR не указано, то эти значения передаются в процедуру.

Изучите пример:



Дополнительно:

1. Три золотых правила.
2. Глобальные и локальные переменные.
3. Формальные и фактические параметры.
4. Параметры-значения и параметры-переменные.

1. Три золотых правила:

1. если надо что-то вычислить и получить в точке вызова результат − используем функцию;
2. если надо что-то сделать: считать данные с диска, очистить часть экрана, закрасить экран в другой цвет и т.п. − используем процедуру;
3. при вызове процедуры или функции тип фактических параметров должен совпадать с типом формальных параметров (т.е. они должны быть согласованы).

2. Глобальные переменные (константы, типы) − переменные (константы, типы), объявленные в программе вне процедуры или функции.

Локальные − существуют только внутри процедуры или функции, объявленные либо в списке параметров, либо в разделе VAR внутри процедуры или функции (для констант − раздел CONST, для типов − TYPE).

При совпадении имен локальной и глобальной переменной используется в подпрограмме локальное имя (если в подпрограмме используется цикл FOR, то параметр цикла − локальная переменная).

3. Формальные параметры − описываются в заголовке объявления подпрограммы.

Фактические параметры − которые подставляются на их место при вызове подпрограммы.

4. Параметры-значения − локальные переменные подпрограммы, стартовые значения которых задаются при вызове подпрограммы из внешних блоков, могут изменять свои значения, но изменения только локальные.

Параметры-переменные − локальные переменные подпрограммы, стартовые значения которых задаются при вызове подпрограммы из внешних блоков, могут изменять свои значения и изменения носят глобальный характер (т.е. передаются в вызывающие операторы).

Пример.

Возведение числа в квадрат, если квадрат числа больше 100, то программа возвращает значение 100.

(pr1− использует функцию, pr2−использует для решения той же задачи прцедуру)

program pr1;

var flag:boolean; {flag-глобальная переменная}

function q(x:real):real; {x-локальная переменная,

 парметр-значение,

 формальный параметр}

const max=100; {локальная константа}

var x2:real; {x2-локальная переменная}

 begin

 x2:=sqr(x);

 flag:=(x2>max);

 if flag then x2:=max;

 q:=x2;

 end;

Begin

 writeln(q(5),'Флаг',flag); {5-фактический параметр}

End.

program pr2;

var flag:boolean; {flag-глобальная переменная}

procedure q(x:real;var sq:real); {x-локальная переменная,

 парметр-значение,

 формальный параметр}

const max=100; {локальная константа}

var x2:real; {x2-локальная переменная}

 begin

 x2:=sqr(x);

 flag:=(x2>max);

 sq:=x2;

 end;

var sqGlobal:real;

Begin

 q(5,sqGlobal); {5-фактический параметр}

 writeln(sqGlobal,'Флаг',flag);

End.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что представляют собой формальные и фактические параметры программы и как организовать их согласование?

2. В какую точку программы осуществляется выход из функции?

3. В чем различие между локальными и глобальными переменными и могут ли совпадать их имена?

4. Назвать два вида локальных переменных и описать их различие.

# ВАРИАНТЫ ЗАДАНИЙ

Задание:

1. Написать программу, содержащую функцию, которая вычисляет объем цилиндра. Параметрами функции должны быть радиус и высота цилиндра.
2. Написать программу, содержащую функцию, которая возвращает процент от числа, полученного в качестве аргумента.
3. Написать программу, содержащую функцию, которая вычисляет факториал числа, введенного с клавиатуры.